I Didn't Know You Could Do That!

Paul Kimpel & Accomplices

2014 Universe Conference

Wednesday, 15 October 2014, 11:00 a.m.

Paradigm Corporation

Generating File Names with UNIQUETOKEN

Did you know – that you can get the MCP to put things like dates, times, and mix numbers in file names automatically via file equation?

Many Ways to Build a File Name

◆ Assigned to the **FILENAME**, **TITLE**, **LFILENAME**, or **LTITLE** file attributes

◆ Construct a string in a program

◆ Use string expressions in WFL file equations

◆ Use DO file parameters in CANDE

— *OR* —

◆ Do it automatically with the **UNIQUETOKEN** file attribute and file name *keywords*

Paradigm   3

What is UNIQUETOKEN?

◆ Defines an escape character within file names

◆ Specified as a single-character string attribute
  • Must be a printable character
  • Must not be a character otherwise used in the name
  • Must be followed in the name by a valid file name keyword – that file name node must be quoted

◆ Restrictions
  • Only for disk files
  • Keyword substitution only occurs when creating a file

◆ Example:
```
RUN OBJECT/MAKER; FILE DATA(NEWFILE,
  UNIQUETOKEN="@",
  TITLE=DATA/"XX@MIXNOYY"/DECK ON PACK)
```

Paradigm   4

File Name Keywords

- ◆ **MIXNO** – task mix number

- ◆ **JOBNO** – task job number

- ◆ **HOSTNAME** – local host name

- ◆ **JDATE** – Julian date (YYYYDDD)

- ◆ **DATE** – Gregorian date (YYYYMMDD)

- ◆ **TIME** – time of day (HHMMSS)

- ◆ **TIMEMIC** – time of day in 2.4 μsec "ticks"
  (as ten digits: 999999999)

- ◆ **SYSTEMSERIAL** – local system serial number

Paradigm

5

References

- ◆ *File Attributes Programming Reference Manual*
  - • FILENAME attribute
  - • UNIQUETOKEN attribute

Paradigm

6

Capturing
Programdumps to Disk

Did you know – that you can avoid wasting paper printing unnecessary programdumps by directing them to disk and analyzing them selectively?

About Programdumps

◆ A program fault-analysis and debugging tool
- Dumps a task's stack and related data areas to a semi-analyzed printer file
- Output when a program DS-es or on program request
- Controlled by the `OPTION` task attribute

◆ Dumping to printer is the original and default output method

◆ There is another way – programdump *to disk*
- Task state is written to a "`PDUMP`" file
- Analyzed by `SYSTEM/DUMPANALYZER` utility
- Can be configured system-wide or at the task level

Paradigm

8

Controlling Programdumps

◆ At the system level
  • Set System Option 18 – **PDTODISK**

◆ At the task level with the **OPTION** attribute
  • Options to control generating a dump
    – **FAULT**, **DSED**
  • Options to override the destination of a dump
    – **TODISK**, **TOPRINTER**
  • Options to control formatting of a dump
    – **ARRAYS**, **BASE**, **CODE**, **CRITICALBLOCK**, **DBS**, **FILES**, **LIBRARIES**, **PDUMPONLY**, **PRESENTARRAYS**, **PRIVATELIBRARIES**
  • See the *Task Attributes Programming Reference Manual* for details

Paradigm

9

PDUMP Files

◆ Created when a dump is created **TODISK**

◆ Contains most of the data needed to analyze the dump later, off-line
  • DA will usually need access to the task's codefile
  • DA may need access to the same MCP codefile and any libraries that are connected to the dumping task

◆ File name (under current usercode and family):
```
PDUMP/
    <codefile name>/
    <date yymmdd>/
    <time hhmmss>/
    <mix number>
```

Paradigm

10

Analyzing a PDUMP File

- ◆ **SYSTEM/DUMPANALYZER** (DA)
  - • Batch and interactive modes
  - • Many commands for interactively examining memory and decoding MCP system structures
  - • Integrated with **OBJECT/ED** for contextual debugging
  - • There's a learning curve…

- ◆ DA has a command that will format a dump almost the same as the MCP programdump
  - • **PROGRAMDUMP**
  - • Options
    - − **ALL, ARRAYS, BASE, CRITICALBLOCK, CODE, DBS, FILES, LIBRARIES, PRIVATELIBRARIES**

Paradigm

11

Sample DUMPANALYZER Job

```
BEGIN JOB PDUMP/ANALYZER;
JOBSUMMARY = SUPPRESSED;
NOJOBSUMMARYIO;
PRINTDEFAULTS = (PRINTDISPOSITION=EOJ,
    DESTINATION="DCMAIL",
        NOTE="DC-EA=PAUL.KIMPEL@DIGM.COM; " &
            "DC-ES=PDUMP;DC-EX=TXT");

RUN *SYSTEM/DUMPANALYZER;
    FILE OPTIONS(READER);
    FILE TAPEIN(DISK, TITLE=
        (TEST)PDUMP/NEW/OBJECT/SOURCE/
        RPMPORTAL/INVOICING/100406/151546/08274
    ); DATA
PROGRAMDUMP ARRAYS
? END JOB
```

Paradigm

12

References

◆ *Task Attributes Programming Reference Manual*

◆ *Task Management Programming Guide*
- Under section 10, "Determining Process History"
- See "Controlling Programdumps"

◆ *System Software Utilities Operations Guide*
- Section 9, "DUMPANALYZER"

◆ Andrew Lesh's video:
- http://youtu.be/RZ9503l1otk

Paradigm

13

END

I Didn't Know You Could Do That!

2014 Universe Conference